CS 2413 Data Structures – Spring 2017 – Programming Project 2
Due March 1, 2017 – 11:59 PM

Objectives

1. [20 pts] Implement Operators = assignment, << ostream; Copy Constructor; Destructor in all the classes that you create.
2. [10 pts] Create vector-like operations when items (item pointers) are added and removed.
3. [50 pts] Design and implement the program *per the project description*. 50 pts are distributed as follows:
   a. [10 pts] Creating the Bag class that has all the necessary fields and methods.
   b. [10 pts] Process the AddItem command to add items corresponding to a user.
   c. [10 pts] Process DelItem command to remove an item from a Bag of a user.
   d. [10 pts] Print out the information of a specific item by going through the Bag of each user.
   e. [10 pts] Summary information about the items in the Bag for each user, including the number of items in each item category, total price for each item category, and total value of all items in the Bag.
4. [20 pts] Document your project thoroughly as the examples in the textbook. This includes but is not limited to header comments for all classes/methods, explanatory comments for each section of code, meaningful variable and method names, and consistent indentation.

Project Description

This project is an extension of the previous project. You will create a new class named Bag. Your Bag class may look like this:

```
class Bag {
        protected:
                Item* myItems;
                int     numItems;
        public:
                // all the methods required
}
```

As you'll notice, myItems is a pointer to an Item object. It can be initialized with the following statement:

```
myItems = new Item[1]; //creating an array of size 1
```

As you add and remove items the value numItems should be changed. Additionally, when you add more items, you will need to resize the myItems array – for example, you can do something like this:

```
Item* temp = myItems;
myItems = new Item[numItems++];
for (int i=0; i < numItems-1; i++) { myItems[i] = temp[i]; temp[i] = NULL; }
delete [] temp;
// put the new item in numItems position and rest of the code.
```

You will need similar code when you delete an item. **Note that you must resize the array as items are added and removed.**

In the input line, after the AddItem the next integer will be a user number. For example, in the line below 1876 will be a user ID. The integer that follows that will be an item number – in the case below, it is 333 (you need to modify to your Item class to have an itemNumber field).

```
AddItem 1876 333 5 1 2 3 7.75 7.62 0.69 0.025 4.97  2   0  10
```

After you read the above line of input, you will check an array of integers declared in the main program as follows:

```
int users[50]; //50 maximum users
int numUsers;
Bag* userBag[50]; //users[i] Bag pointer is in userBag[i]
```

If the user ID 333 is found in location 5 of the users array, for example, then the Bag object corresponding to this user is *Bag[5]. If the user ID is not found, then we add the new user ID at the end of the array (in position numUsers and increment numUsers) and create a new Bag object and place the pointer to it in position UserBag[numUsers]. Make sure to increment numUsers after a new user is added.

In addition to the AddItem command you will also have a DelItem command. For example, to remove the item numbered 333 from the user with ID 1876, the command will be as follows:

```
DelItem 1876 333
```

All other requirement for the project will be like Project 1. Please read the requirement d and e above to print item and summary information.

Constraints

1. In this project, the only header you will use is #include <iostream>.
2. None of the projects is a group project. Consulting with other members of this class on programming projects is strictly not allowed and plagiarism charges will be imposed on students who do not follow this.